assignment-4

October 30, 2024

[1]:
```python
import sqlite3
from flask import Flask, render_template, request, redirect, url_for
from IPython.display import IFrame
import threading

DATABASE = 'tasks.db'

# Initialize the database
def init_db():
    conn = sqlite3.connect(DATABASE)
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS tasks (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            description TEXT NOT NULL,
            status INTEGER DEFAULT 0
        )
    ''')
    conn.commit()
    conn.close()

init_db()
```

[2]:
```python
app = Flask(__name__)

# Function to connect to database
def get_db_connection():
    conn = sqlite3.connect(DATABASE)
    conn.row_factory = sqlite3.Row
    return conn

# Routes
@app.route('/')
def home():
    conn = get_db_connection()
    tasks = conn.execute('SELECT * FROM tasks').fetchall()
    conn.close()
    return render_template('home.html', tasks=tasks)

@app.route('/create', methods=('GET', 'POST'))
def create():
    if request.method == 'POST':
        description = request.form['description']
        conn = get_db_connection()
        conn.execute('INSERT INTO tasks (description) VALUES (?)',␣
 ↪(description,))
        conn.commit()
        conn.close()
        return redirect(url_for('home'))
    return render_template('create.html')

@app.route('/update/<int:id>', methods=('GET', 'POST'))
def update(id):
    conn = get_db_connection()
    task = conn.execute('SELECT * FROM tasks WHERE id = ?', (id,)).fetchone()

    if request.method == 'POST':
        description = request.form['description']
        status = request.form.get('status', 0)
        conn.execute('UPDATE tasks SET description = ?, status = ? WHERE id = ?
 ↪',
                     (description, int(status), id))
        conn.commit()
        conn.close()
        return redirect(url_for('home'))

    conn.close()
    return render_template('update.html', task=task)

@app.route('/delete/<int:id>', methods=('POST',))
def delete(id):
    conn = get_db_connection()
    conn.execute('DELETE FROM tasks WHERE id = ?', (id,))
    conn.commit()
    conn.close()
    return redirect(url_for('home'))

# Run the app in a separate thread
def run_app():
    app.run(debug=True, use_reloader=False)

# Start the Flask app in a thread
thread = threading.Thread(target=run_app)
thread.start()
```

```
 * Serving Flask app '__main__'
 * Debug mode: on
```

```
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [30/Oct/2024 20:23:42] "GET / HTTP/1.1" 200 -
```

[3]:
```
%%writefile templates/home.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task Manager</title>
</head>
<body>
    <h1>Task Manager</h1>
    <a href="{{ url_for('create') }}">Add New Task</a>
    <ul>
        {% for task in tasks %}
        <li>
            {{ task['description'] }} -
            {% if task['status'] == 1 %} Complete {% else %} Incomplete {%␣
 ↪endif %}
            <a href="{{ url_for('update', id=task['id']) }}">Edit</a>
            <form action="{{ url_for('delete', id=task['id']) }}" method="post"␣
 ↪style="display:inline;">
                <button type="submit">Delete</button>
            </form>
        </li>
        {% endfor %}
    </ul>
</body>
</html>
```

```
Overwriting templates/home.html
```

[4]:
```
%%writefile templates/create.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Create Task</title>
</head>
<body>
    <h1>Create a New Task</h1>
    <form action="{{ url_for('create') }}" method="post">
```

```
        <label for="description">Task Description:</label>
        <input type="text" name="description" id="description" required>
        <button type="submit">Create Task</button>
    </form>
    <a href="{{ url_for('home') }}">Back to Task List</a>
</body>
</html>
```

```
Overwriting templates/create.html
```

[5]:
```
%%writefile templates/update.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Update Task</title>
</head>
<body>
    <h1>Update Task</h1>
    <form action="{{ url_for('update', id=task['id']) }}" method="post">
        <label for="description">Task Description:</label>
        <input type="text" name="description" id="description" value="{{␣
 ↪task['description'] }}" required>

        <label for="status">Complete:</label>
        <input type="checkbox" name="status" id="status" value="1" {% if␣
 ↪task['status'] == 1 %}checked{% endif %}>

        <button type="submit">Update Task</button>
    </form>
    <a href="{{ url_for('home') }}">Back to Task List</a>
</body>
</html>
```

```
Overwriting templates/update.html
```

[6]:
```
IFrame(src="http://127.0.0.1:5000/", width=700, height=400)
```

[6]:
```
<IPython.lib.display.IFrame at 0x20efc6447d0>
```